Page No.
Date

Tutorial Sheet-1

① int a=0, b=0;
for (i=0; i<n; i++)   a+ = rand(·);
for (j=0; j<m; j++)   b+ = rand();

Time Complexity = O(n+m);
Space Complexity = O(1)   Ans.

② int sum = 0, i;
for (i=0; i<n; i=i+2) { Sum+=i; }
↳ loop will run even times.

∴ 0+2+4 ___ 2n
= O(n/2)
⟹ Time Complexity = O(n) Ans.

③ int sum=0, i;
for (i=0; i<n; i=i*2)   { Sum+=i; }

Time Complexity = O(log n) Ans.

④ int sum = 0, i;
for (i=0; i*i<n; i++)

O(log n) Ans.

⑤ int j=1, i=0;
while (i<=n) {
i = i+j;
j++;
}

Time = O(n)
Space = O(1)   Ans.

$\longrightarrow T(n)$

Page No.
Date

⑥. void recurssion (int n) {
if (n == 1) return; $\rightarrow T(1)$
recurssion (n-1); $\rightarrow T(n-1)$
print (n); $\rightarrow T(1)$
recurssion(n-1); $\rightarrow T(n-1)$
}

$\boxed{T(n) = T(n-1) + 1}$ Ans

$T(n-1) = T(n-2) + 1$
$T(n) = T(n-2) + 2$ $\boxed{T(1) = 1}$

$T(n) = T(n-k) + k$
$\hookrightarrow O(n)$ Ans.

⑦. The given recursive funcⁿ is of binary search.

$\therefore$ Time Complexity $= \boxed{T\left(\frac{n}{2} + 1\right)}$

Or

n is dividing by 2 every time
$\therefore O(\log n)$ Ans.

⑧. (i) $T(n) = T(n-1) + 1$
$T(n-1) = T(n-1) + 1$
$T(n) = T(n-2) + 2$
$T(n-1) = T(n-k) + K$
$= O(n)$ Ans.

ii) $T(n) = T(n-1) + n$
$T(n-1) = T(n-2) + (n-1)$
$T(n) = T(n-2) + 2n - 1$
$T(n-2) = T(n-4) + 2n - 5$

Scanned with CamScanner

Page No.
Date

$T(n) = T(n-4) + 4n - 6$

$T(n) = T(n-K) + Kn - (K-1)(2)$

iii) $T(n) = T(n/2) + 1$

$a = 1,\ b = 2,\ K = 0,\ p = 0$

$a = b^K$

$O(n^{\log_2 1} \log^{p+1} n)$

$O(\log n)$ Ans.

10) $T(n) = 2T(n/2) + 1$

$a = 2,\ b = 1,\ K = 0$

$2 > 1,\ (a > b^K)$

$O(n^{\log_2 1}) = O(n)$ Ans.

(v) $T(n) = 3T(n-1)$

$T(0) = 1$

$T(n-1) = 3T(n-2)$

$T(n) = 3(3T(n-2))$

$T(n) = 3^K T(n-K)$

$n - K = 0$

$\boxed{n = K}$

$3^n \times 1 = T(n)$

$T(n) = 3^n$

$\boxed{O(3^n)}$ Ans.

9. $O(n)$

10. $O(N + N)$

11. $O(N \log N)$

12. X will be always a better choice for large inputs.

13. $O(\log N)$

14. $T(n) = 7T(n/2) + 3n^2 + 2$

using Master's Theorem

$a = 7,\ b = 2,\ K = 2$

$a > b^K$

$7 > 2^2$ = True

$T(n) = O(n^{\log_2 7})$

$= O(n\hat{\ }2.8)$ Ans.

15. $f1(n) = n\hat{\ }\sqrt{n}$

$f2(n) = 2\hat{\ }n$

$f3(n) = (1.00001)\hat{\ }n$

$f4(n) = n\hat{\ }(1.0 + 2\hat{\ }(n/2))$

~~f(4)~~ $f4 > f2 > f3 > f1$

Page No.
Date

(16). $f(n) = 2^{(2n)}$

$\Omega(2^n)$

(17). $T(n) = 2T(n/2) + n^2$

$a = 2, \quad b = 2, \quad k = 2$

$2 < 2^2 \quad \boxed{p = 0}$

$T(n) = \Theta(n^k \log^p n)$

$T(n) = \Theta(n^2)$ Ans.

(18).
```
int gcd (int n, int m) {
  if (n % m == 0) return m;
  if (n < m) swap (n, m);
  while (m > 0) { n = n % m;
                  swap (n, m); }
  return m; }
```
↳ here n is ~~gene~~ gradually decreasing

$\therefore O(\log n)$ Ans.

(19).
```
int a = 0, b = 0;                  → O(n)
for (i = 0; i < N; i++) {
    for (j = 0; j < N; j++)        → O(n)
        a = a + j;
}                                    O(n^2)
for (K = 0; K < N; K++) {
    b = b + n;                     —  O(n)
}
```

$O(n \cancel{*} n + n)$

$O(n^2 + n)$

$= O(n^2)$ Ans.

Scanned with CamScanner